



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,499	03/24/2004	David John Butcher	550-540	4256
23117 7590 10/18/2010 NIXON & VANDERHYE, PC 901 NORTH GLEBE ROAD, 11TH FLOOR ARLINGTON, VA 22203				
EXAMINER				
LI, AIMEE J				
ART UNIT		PAPER NUMBER		
2183				
MAIL DATE		DELIVERY MODE		
10/18/2010		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

1                                   RECORD OF ORAL HEARING  
2                   UNITED STATES PATENT AND TRADEMARK OFFICE  
3                                   \_\_\_\_\_  
4                   BEFORE THE BOARD OF PATENT APPEALS  
5                   AND INTERFERENCES  
6                                   \_\_\_\_\_

7                   *Ex Parte* DAVID JOHN BUTCHER, STEPHEN JOHN HILL,  
8                   and WILCO DIJKSTRA  
9                                   \_\_\_\_\_

10                                   Appeal 2009-010086  
11                                   Application 10/807,499  
12                                   Technology Center 2100  
13                                   \_\_\_\_\_

14                                   Oral Hearing Held: August 11, 2010  
15                                   \_\_\_\_\_

16                   Before JOSEPH L. DIXON, STEPHEN C. SIU, and JAMES R. HUGHES,  
17                   *Administrative Patent Judges.*

18                   APPEARANCES:

19                   ON BEHALF OF THE APPELLANT:  
20  
21

22                   STANLEY C. SPOONER, ESQUIRE  
23                   Nixon & Vanderhye, P.C.  
24                   901 N Glebe Rd., 11th Floor  
25                   Arlington, Virginia 22203  
26

1           The above-entitled matter came on for hearing on Wednesday, August  
2 11, 2010, commencing at 11:22 a.m., at the U.S. Patent and Trademark  
3 Office, 600 Dulany Street, Alexandria, Virginia, before Timothy J.  
4 Atkinson, Jr., a Notary Public.

5           THE USHER: Calendar No. 32, Appeal No. 2009-010086,  
6 Mr. Spooner.

7           JUDGE DIXON: Hello, Mr. Spooner. Welcome back.

8           MR. SPOONER: Yes, I guess.

9           JUDGE DIXON: Get a double on Nixon and Vanderhye today.

10          MR. SPOONER: Okay. Long morning.

11          JUDGE DIXON: Yep. Sorry for keeping you waiting.

12          MR. SPOONER: What? Not a problem. Not a problem.

13          JUDGE DIXON: Everybody likes to talk today, I guess.

14          MR. SPOONER: I guess. Well --

15          JUDGE DIXON: Or maybe we do.

16          MR. SPOONER: Well, I have a way to shortcut this, but --

17          JUDGE DIXON: Excellent. Excellent.

18          MR. SPOONER: -- I'll try it out on you.

19          JUDGE DIXON: Okay.

20          MR. SPOONER: As you know, I'm Stanley Spooner, representing the  
21 Assignee of record, Arm Limited. It's a U.K. company. And I want to hit a  
22 couple high points from the background of the invention, and this is between  
23 pages 2, line 16 to page 3, line 29, and they explain what the difference is  
24 between this invention and the prior art. So, it's known to use -- Arm does  
25 processors which have -- which support two instruction sets. The arm  
26

1 instruction set, which are 32-bit instructions, and the thumb instruction set,  
2 which are compressed into 16-bit instructions.

3 It's also known that Arm makes processors that can easily switch  
4 between a java state, in which java bite codes are treated as native  
5 instructions, and the arm/thumb state. There's a problem switching back and  
6 forth, and one known solution to switching is to do what the prior art does.  
7 Use a single instruction in which a comparison and branch routine switches  
8 program execution from one point to another are combined. In other words,  
9 a known compare and branch instruction calculates a target branch using a  
10 field within the instruction itself, and that's in the spec at page 2, lines 32 to  
11 34. Obviously, if you're using a field in the instruction, you're limited in the  
12 size of that field, and that provides a very limited flexibility, and that's  
13 discussed on page 3, lines 2 to 4.

14 The present inventions solves that problem by providing a compare  
15 and branch instruction. Same overall general instruction, but one that uses --  
16 that determines the target branch address from a predetermined -- from a  
17 preprogrammed storage value. In other words, that value is stored  
18 somewhere else. It's not in the instruction. So, we don't go back to the  
19 instruction. We go somewhere else and the decoder in the claims sets out  
20 how you find that somewhere else. All right. So, the issue is, does a  
21 compare and branch instruction that carries the address within the instruction  
22 render obvious a compare and branch instruction that determines the address  
23 from a preprogrammed stored valued and a program counter value, which is  
24 what the claim does. All right.

1           The claims briefly -- independent claims 1, 13, 25 and 37; they all  
2   recite either a decoder, having certain functionality or a series of function  
3   steps in the method claims. Turning right to the Appeal Brief, in section A,  
4   the Examiner admits that Ishizaki doesn't teach the copying and determining  
5   steps, and we've argued in the brief that she also makes admissions that  
6   suggest that it doesn't teach the branching step, or the branching function of  
7   the decoder. However, in the Examiner's answer, she then comes in and  
8   suggests that the copying step is somewhere disclosed in Ishizaki's abstract.  
9   We've gone through it. I think the Reply Brief notes that it's just not there.  
10   An important part of the -- and the Reply Brief hasn't been traversed as far  
11   as I know by the Examiner in the supplemental or anything like that.

12           If we go to the Appeal Brief, section B, that's the section which noted  
13   that the Hennessy reference doesn't do -- she's alleged that Hennessy teaches  
14   the copying step that she previously suggested was missing from Ishizaki,  
15   but we noted that Hennessy's copying is in response to an exception  
16   occurring, not in response to the comparison result. So, she's not done this  
17   comparison and she's not shown how this Hennessy response is in response  
18   to that. It's in response to an exception. That's clearly stated in Hennessy,  
19   page 411, lines 22 to 25. So, in the obviousness rejection, the Examiner's  
20   got the obligation first to point out where the claimed elements, and where  
21   the claimed interrelationships are disclosed in amongst the combination of  
22   references. In this instance, she's pointed to where structures are shown and  
23   certain functions are shown, but she hasn't shown the interrelationship  
24   between those functions.

1 And, if I can digress slightly, I think this case is almost identical to an  
2 earlier Board case that I had. Appealing Examiner rejection, and that was  
3 Appeal No. 2009-005640, decided January 26th of this year, and that's a  
4 case in which Judge Hughes pointed out that the -- well, I'm reviewing --

5 JUDGE HUGHES: Go ahead.

6 MR. SPOONER: -- this case and I'm saying, wait a minute I just had  
7 this same issue. And so, that's how I think we can shortcut this unless the  
8 Board has any serious questions, because in that exact case the Board held  
9 that, while the prior art does teach processors, decoders, null value checks  
10 and null value exception handlers, as recited in claim 1. The Board properly  
11 went further and found that the prior art did not teach a decoder that  
12 performs both functions - performing a null value check and controlling  
13 branching to a null value exception handler. So that's -- I think that's the  
14 situation we have here. There are bits and pieces of the claim that are  
15 sprinkled in the two cited prior art references, but there is no  
16 interrelationship as set out in the claims.

17 Now what I think may have contributed to some of the Examiner's  
18 confusion here is that -- wait a minute. I missed it. Oh, it's the difference in  
19 the meaning of the word exception. Now we mentioned this in the Reply  
20 Brief, because that's when I sort of realized it. The Hennessy references  
21 states that exceptions are events other than branches or jumps, and that's at  
22 page 10, section 5.6, line 4. The Examiner says that Ishizaki teaches that an  
23 exception is a conditional branch. That's the Examiner's answer at page 16,  
24 lines 4 to 5. Well, if the exception isn't conditional branch, and Hennessy  
25 says that an exception is events other than a branch, they don't mesh. So, the

26

1 exception in one is not the same thing as an exception in the other. And so,  
2 how the Examiner marries the two is simply not apparent.

3         The second -- so, the first prong of the obviousness is all the claimed  
4 elements and interrelationship between elements aren't disclosed in the prior  
5 art reference combination. Even if they were the Examiner doesn't meet her  
6 burden of providing a clear explicit rationale as to why she's picking and  
7 choosing bits from the references, and then combining them in the manner of  
8 the claims. We treated that in the Appeal Brief. I don't think we have to  
9 belabor it.

10         Turning to a couple points that were visited in the Reply Brief, error  
11 number 2, I think we discussed. The Examiner now is citing some  
12 secondary references as teaching the copying step when she didn't  
13 previously do that. We couldn't find it, and the Reply Brief goes through  
14 that. We couldn't find that in the teaching anywhere. We discussed her  
15 failure to provide a motivation in error number 3. We also discussed error  
16 number 4, the precedent for the fact that if one of the prior art combinations  
17 teaches away from the invention, that combination can't possibly teach the  
18 invention. In other words, it rebuts any prima facie case that was made even  
19 if there was one made. So in sum, hearing no questions, in summary we've  
20 got no prima facie case of obviousness, because the first prong is not met.  
21 The claimed interrelationships between the decoder functions are simply not  
22 mentioned. The second prong, there's no analysis of her reasons for picking  
23 and choosing bits from the two references. So, there is no prima facie case  
24 of obviousness, but, even if there was, the whole point of the fact that  
25 Ishizaki teaches at column 5, lines 51 to 57, that instructions dot space, dot  
26

1 space, dot space, dot, are examined to detect the type of exception that has  
2 occurred. That teaches the examination of the instruction, which is exactly  
3 what the prior art teaches for these compare and branch instructions. You  
4 look at the instruction for what's there. That's not what we claim. That  
5 would preclude what we're doing, because if you went to the instruction you  
6 wouldn't be doing the other stuff that we require. So, no prima facie case for  
7 either of the first or second prongs, and, even if there was one it's rebutted  
8 by the Ishizaki reference.

9 JUDGE HUGHES: What was that cite you just --

10 MR. SPOONER: The --

11 JUDGE HUGHES: -- for your quote?

12 MR. SPOONER: -- Ishizaki is at column 5, lines 51 to 57. It says,  
13 instructions and then it has some intervening --

14 JUDGE HUGHES: Right

15 MR. SPOONER: -- parenthetical inspection -- insertion, but then it  
16 says, instructions are examined to detect the type of exception that has  
17 occurred. That's the characteristic of the compare and branch instruction  
18 that we talked about in the background of the invention that I mentioned  
19 when we first went through it.

20 JUDGE HUGHES: I'm going to make you work for this a little bit,  
21 but --

22 MR. SPOONER: Okay.

23 JUDGE HUGHES: All right. Starting with claim 1, you've got an --  
24 this is an apparatus claim --

25 MR. SPOONER: Yes, sir.

26



1 JUDGE HUGHES: -- from the preamble, but you've got a lot of looks  
2 like method steps in here. So, you've got a lot of functional language.

3 MR. SPOONER: Right.

4 JUDGE HUGHES: How does that, the functional language,  
5 distinguish this apparatus from prior art?

6 MR. SPOONER: Well, it's functional language, but it's functional  
7 language directed to the decoder. As you may appreciate, there are about  
8 eight gazillion ways to put together a decoder, and -- so, the specific circuit  
9 in that decoder that does a particular function, that goes a particular place,  
10 that stores a particular comparison result, retrieves it -- those circuits aren't  
11 significant. It's the whole idea of a decoder, which does -- which is  
12 programmed, or created, or hardwired, or a combination of program and  
13 hardware that do these functional interrelationships. And so, the functional  
14 language defines what the structure of that decoder has to be. And there are,  
15 like I say, many different ways to skin that particular cat. But the end result  
16 is you have to have a decoder that does those things. And that 's the  
17 interrelationship between those things that the decoder does, is what  
18 distinguishes claim 1 from the Ishizaki reference.

19 JUDGE HUGHES: As far as you're performing a comparison, in  
20 Ishizaki -- you're talking about performing a comparison between two  
21 registers for doing an exception check. Is that correct -- in general, the  
22 portion that's cited by the Examiner.

23 MR. SPOONER: Well, we're not disputing that Ishizaki teaches a  
24 comparison.

25  
26

1 JUDGE HUGHES: Right. No, but it's a comparison for an exception  
2 check. Is that correct?

3 MR. SPOONER: I don't think that the claim says exception check.

4 JUDGE HUGHES: Well, I'm just asking --

5 MR. SPOONER: Ishizaki does.

6 JUDGE HUGHES: -- you're characterization of Ishizaki.

7 MR. SPOONER: Ishizaki does. Yes, sir.

8 JUDGE HUGHES: All right.

9 MR. SPOONER: I believe so.

10 JUDGE HUGHES: So then doesn't Hennessy also teach doing  
11 exception checks?

12 MR. SPOONER: Oh yes. Hennessy's limited to exception checks.

13 JUDGE HUGHES: All right. And as far as Hennessy goes, you've  
14 got multiple registers -- receiving the program counter into an exception  
15 program counter. That's disclosed by Hennessy, is it not?

16 MR. SPOONER: That I don't know, but I'll accept your argument that  
17 it is.

18 JUDGE HUGHES: Okay.

19 MR. SPOONER: But the important thing to remember is Hennessy  
20 doesn't do his copying in response to a comparison result. He does it in  
21 response to an exception occurring and the comparison result is --

22 JUDGE HUGHES: Right, based on an exception check --

23 MR. SPOONER: -- different from an exception.

24 JUDGE HUGHES: -- which is the point I'm trying to make.

25 Hennessy does its saving from the program counter to the exception program

26

1 counter based on an exception, which is found by doing an exception check,  
2 which Ishizaki teaches can be a comparison between two registers. I just  
3 want to point out that your language -- the language you're using in the claim  
4 doesn't necessarily exclude Hennessy, based on copying from one register to  
5 another. You're trying to add something in there that isn't necessarily in the  
6 claim.

7 MR. SPOONER: Right, but again the comparison is done between a  
8 value stored in the first register and a value stored in the second register. It's  
9 not part of the instruction. The copying is done in dependence upon that  
10 comparison of things stored in two registers, and that program counter value  
11 is then stored in a third register. It has nothing to do with looking at the  
12 instruction that's starts this thing going off. And then we have the  
13 determination of the branch address based on the preprogrammed stored  
14 value and the program counter value. That's not there.

15 JUDGE HUGHES: I also wanted to ask you about Hennessy  
16 disclosing its -- it discloses a cause register, or a vector address depending  
17 on what system you're using for determining what kind of exception is going  
18 on?

19 MR. SPOONER: Where is this?

20 JUDGE HUGHES: That would be at 412. How exceptions are  
21 handled. Let's see -- yeah, 412. You've got a cause register. It's --

22 MR. SPOONER: Oh, under cause?

23 JUDGE HUGHES: There's a bullet point there.

24 MR. SPOONER: Under cause, the bullet point?

1 JUDGE HUGHES: Yeah. And they also discuss above that -- if  
2 you're doing a vector system for handling exceptions -- a vector address,  
3 you're -- there's a little table there.

4 MR. SPOONER: Um-hum.

5 JUDGE HUGHES: So, aren't those values apart from the instruction  
6 itself? The branching instruction that we're talking about through doing an  
7 exception handling?

8 MR. SPOONER: But it's -- from just my cursory review, is there any  
9 discussion that the register used to record the cause of the exception stores  
10 first and second values, or a program counter value. In other words, are you  
11 trying to read that register -- which of the three claimed registers are you  
12 trying to read that onto?

13 JUDGE HUGHES: Right. I'm not trying to read it onto any of them.

14 MR. SPOONER: Oh, okay. I thought you were, I'm sorry.

15 JUDGE HUGHES: No. The claim language -- let me pull up -- the  
16 Examiner basically cites this information for the determining step. So,  
17 you've got a -- requiring that you determining from a branch address -- or,  
18 excuse me, determining --

19 MR. SPOONER: That's end thing.

20 JUDGE HUGHES: -- a branch address from a preprogrammed stored  
21 value and said program countervalue.

22 MR. SPOONER: Right.

23 JUDGE HUGHES: So, some value in addition to the program value  
24 that's been stored --

25 MR. SPOONER: Right.

26

1 JUDGE HUGHES: -- and the Examiner's pointing to these values as  
2 used in that determining step. That's the way I understood the Examiner's  
3 argument.

4 MR. SPOONER: All right. I'm not sure that the vectored interrupts is  
5 what we're talking about here, but --

6 JUDGE HUGHES: No. I'm just pointing out --

7 MR. SPOONER: Yeah.

8 JUDGE HUGHES: -- that you've got -- well, that's just one -- the  
9 Examiner doesn't clarify exactly what is used for the determining step, but  
10 points to page 412, and says that you've got values there.

11 MR. SPOONER: There are values there.

12 JUDGE HUGHES: So that's the point I was trying to make. You've  
13 got -- let's take, for example, there's a discussion a little farther down on 412  
14 of adding this hexadecimal value for exception handling, and that it is used  
15 with a multiplexer to branch to the exception handling address. The area it  
16 says, C000000X.

17 MR. SPOONER: Sub-hex. Yeah.

18 JUDGE HUGHES: So, that's would be -- that could considered as  
19 stored value, could it not, since it's hardwired into a multiplexer?

20 MR. SPOONER: Oh, sure.

21 JUDGE HUGHES: Okay. I just wanted to clarify. I don't have any  
22 other further questions. Do you guys?

23 MR. SPOONER: But again, to summarize -- make sure we keep the  
24 focus here --

25 JUDGE HUGHES: I understand.

26

1           MR. SPOONER: -- I think that it says there -- the earlier cite to the  
2 register, up under the bullet point cause, was recording the cause of the  
3 exception. So, that's recording some other sort of data. I'm not sure that that  
4 falls within the scope of a program counter value, or that it falls within the  
5 scope of a pre-hyphen programmed stored value. And those are the two  
6 things that are used to determine the target branch address, in accordance  
7 with the claimed decoder. And I just don't see the connection between those  
8 so, and maybe you didn't either, but --

9           JUDGE HUGHES: Well, I'm just trying to clarify where the  
10 Examiner was coming from and see if you understood the argument, so I  
11 think you've covered it.

12          MR. SPOONER: I'm not sure I did --

13          JUDGE HUGHES: All right.

14          MR. SPOONER: -- but anything else?

15          JUDGE DIXON: No questions?

16          MR. SPOONER: Great.

17          JUDGE DIXON: Thank you for your time.

18          JUDGE HUGHES: Thank you.

19          MR. SPOONER: Thank you.

20          Whereupon, the proceedings, at 11:44 a.m., were concluded.

21

22

23

24

25

26